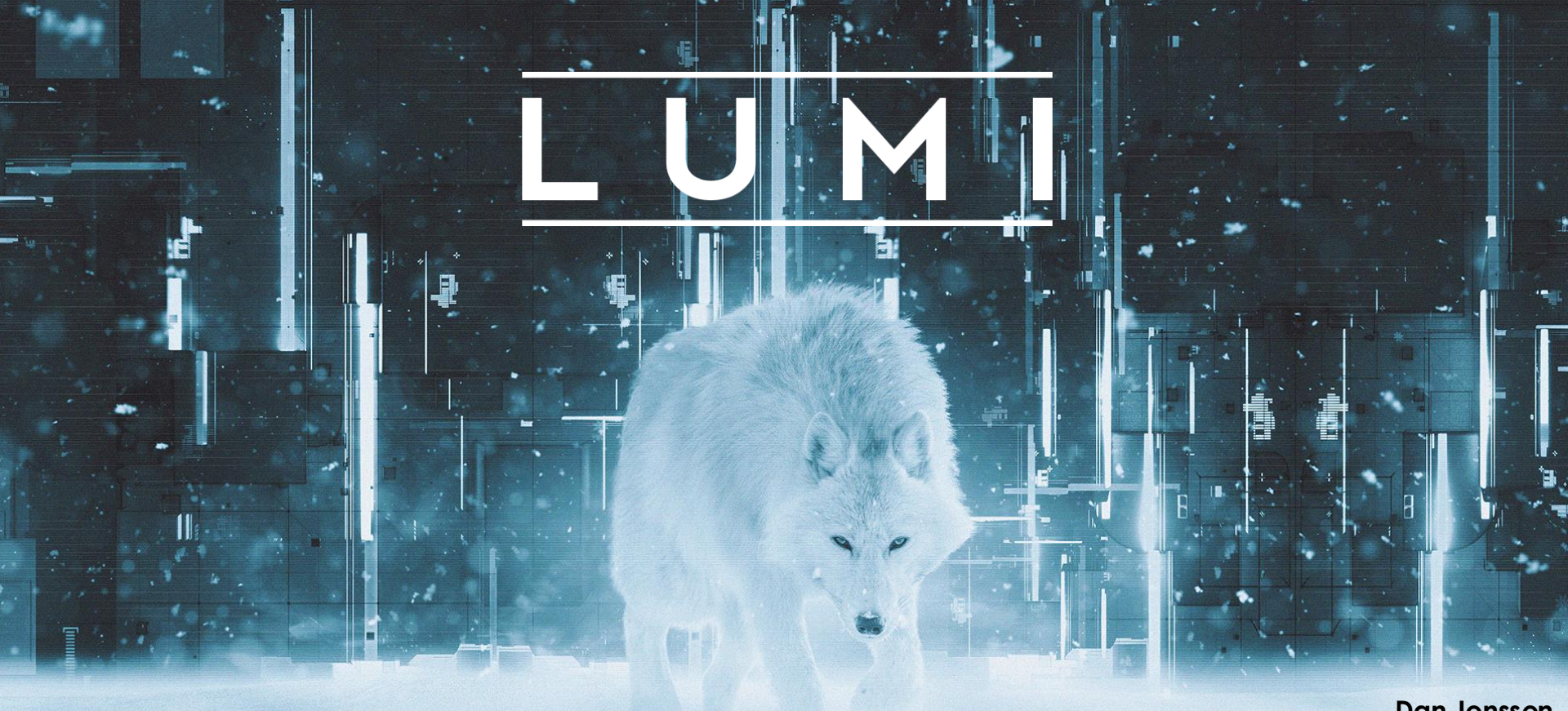


Files on LUMI 2: LUMI-O object storage

Dan Jonsson
Kurt Lust
LUMI User Support Team (LUST)

November 2026

LUMI

Why do I need to know this?

- LUMI-O is the primary option on LUMI to transfer large amounts of data to LUMI
- LUMI-O is the only local option if you want to backup some data
- Some datasets come in a format optimised for object storage rather than a parallel file system

- What we will discuss:
  - Properties of object storage
  - Getting started
  - But not a reference manual of the tools that can be used on LUMI-O

What is LUMI-O?

- Object storage system, based on Ceph
  - Finnish users: similar to Allas, but less functionality at the moment
  - Specific tools to access data, not mounted as a regular file system
- Organisation: 3 levels:
  - Projects: LUMI projects
  - Buckets: "Containers" used to store objects.
    - Flat structure: Buckets cannot contain other buckets
  - Objects: Any type of data, stored in a bucket
    - Atomic access to objects: Put, get, copy, delete, …, but no partial write
    - Flat space, though slashes in the name can create a pseudo-folder view in some tools
  - Metadata for buckets and objects
    - Bucket: e.g., access rights
    - Custom metadata possible

What is LUMI-O? (2)

- Objects can be served on the web also
  - This is how recordings of some LUST courses are served
  - But not meant as a data publishing service (e.g., no EUDAT alternative)
- Can be reached easily from outside LUMI
  - So also a mechanism for data exchange
  - Tools for accessing object storage are more performant and more robust than sftp on high latency connections
- Specs:
  - Capacity: 30 PB
  - Quota: 150 TB capacity, 1K buckets and 500K objects per bucket
  - Billed at 0.25 TB·hour per TB per hour (0.5 before 2025)
  - Persistent for the duration of the project

LUMI

Lustre vs LUMI-O (1)

Lustre	LUMI-O object storage
Closely integrated with compute nodes	Separate system
Upgrades with the system	Separate upgrade cycle and on-the-fly
Organisation: Hierachical directory structure and files	Organisation: Triple flat space of projects, buckets and objects
Files can be read, written, modified, appended, …	Simple atomic operations on objects: put, get, copy, delete
Optimised for bandwidth to the compute nodes	Optimised for reliability, resilience, global access
Simpler schemes for redundancy	Very complex internal redundancy setup

LUMI

Lustre vs LUMI-O (2)

Lustre	LUMI-O object storage
Integrated in the authentication of the supercomputer	Separate key-based authentication mechanism
Seen as any other POSIX file system	Separate range of access tools/APIs, some tools can provide a filesystem view
No external access	External access integrated, includes web
Structure with MDS and ODS	Structure with MDS and ODS, but very different technologies
Parallelism for performance: Access a file in parallel from multiple processes	Parallelism for performance: Different processes access different objects
Fairly expensive to very expensive hardware	Cheaper hardware

LUMI

LUSTRE vs LUMI-O (3)

- The optimal way/technology of storing data is very different depending on whether you work from a parallel file system or from object storage. E.g., in earth and climate science:
  - netCDF is a popular data storage format for storing simulation data on a parallel filesystem
    - Not suited for object storage though as it would be a single object
  - Zarr is a format to store similar data on object storage (cloud storage)
    - It is not a single object, but a structured collection of objects
    - Putting it on a parallel filesystem where each object would become a file in a directory tree is a very bad idea!
    - But with the right libraries, you can access Zarr data on object storage directly from your application

Accessing LUMI-O

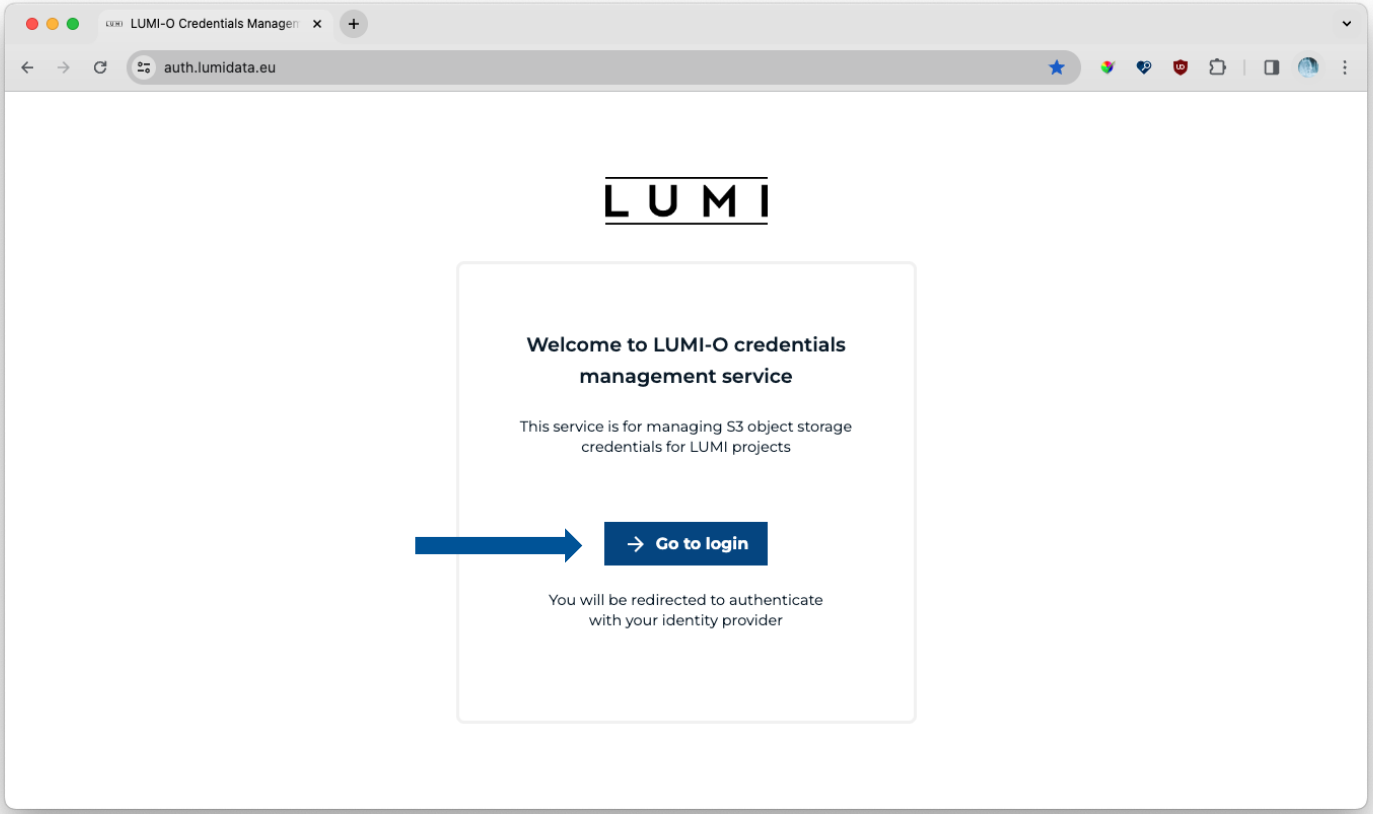
- Access is based on authentication keys
  - Generated via a web interface: Separate steps to generate the credentials and to get them on LUMI.
  - Or generated via Open OnDemand: Can put credentials on LUMI for `rclone` and `s3cmd`.
- Tools:
  - `rclone`: Easiest tool if you want public and private data
  - `s3cmd`
  - `restic`: More a backup tool

  module `lumio`
  - `boto3`: Python API from the AWS SDK for programmatic access
    - Needs a more recent Python version than the system Python
  - Additional GUI-based tools exist for clients
  - Open OnDemand web interface is not a substitute!
    - Speed limited by browser protocols

The credential management web interface

- Credential management web interface at auth.lumidata.eu
  - Create keys
  - Extend lifetime of a key (not needed that frequently anymore)
  - Create configure scripts for various tools

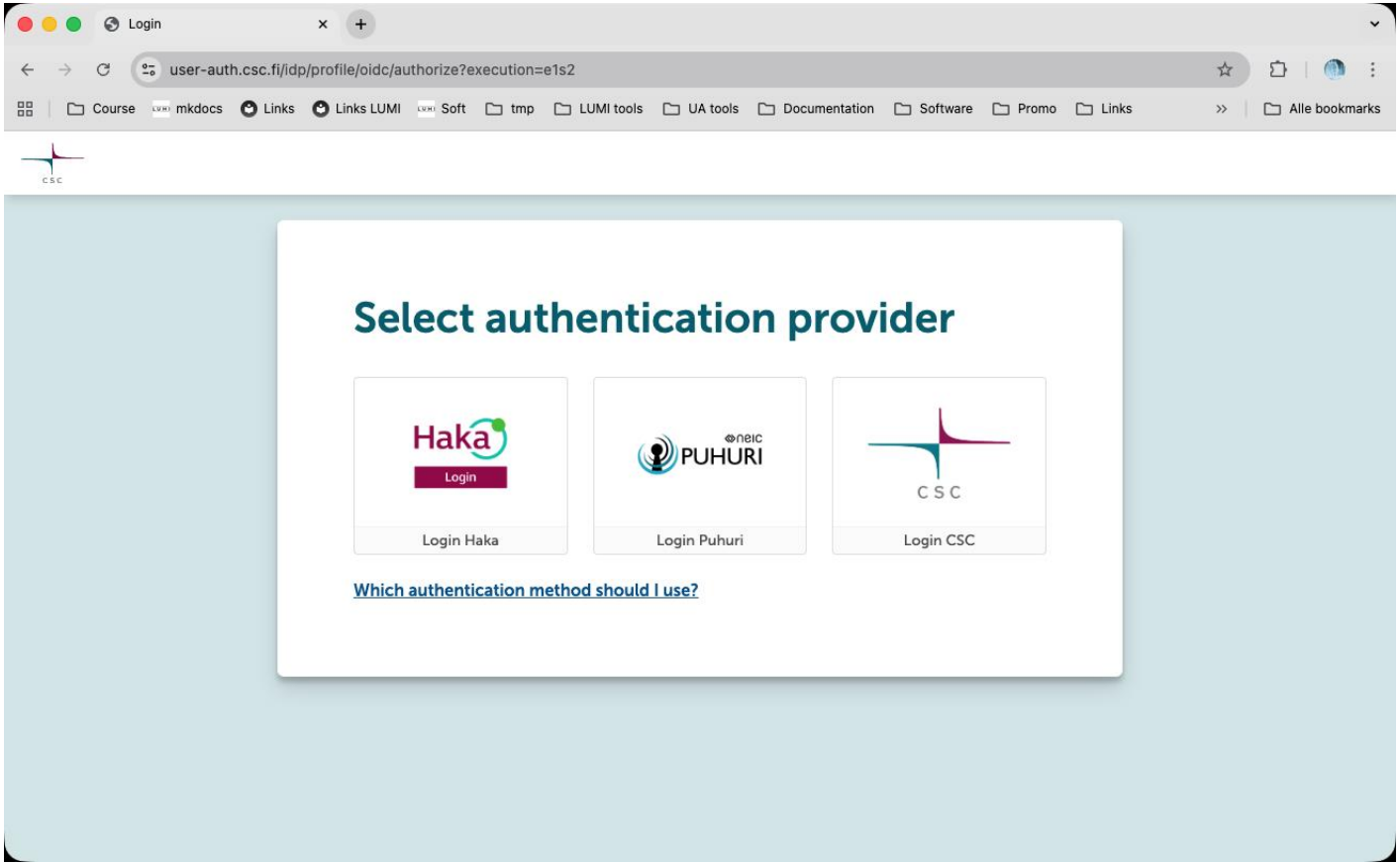
Credential mangement web interface: Create credentials (1)



The credential management web interface

- Credential management web interface at auth.lumidata.eu
  - Create keys
  - Extend lifetime of a key (not needed that frequently anymore)
  - Create configure scripts for various tools
- You'll have to select your login method in the same way as for Open OnDemand

Credential mangement web interface: Create credentials (2)

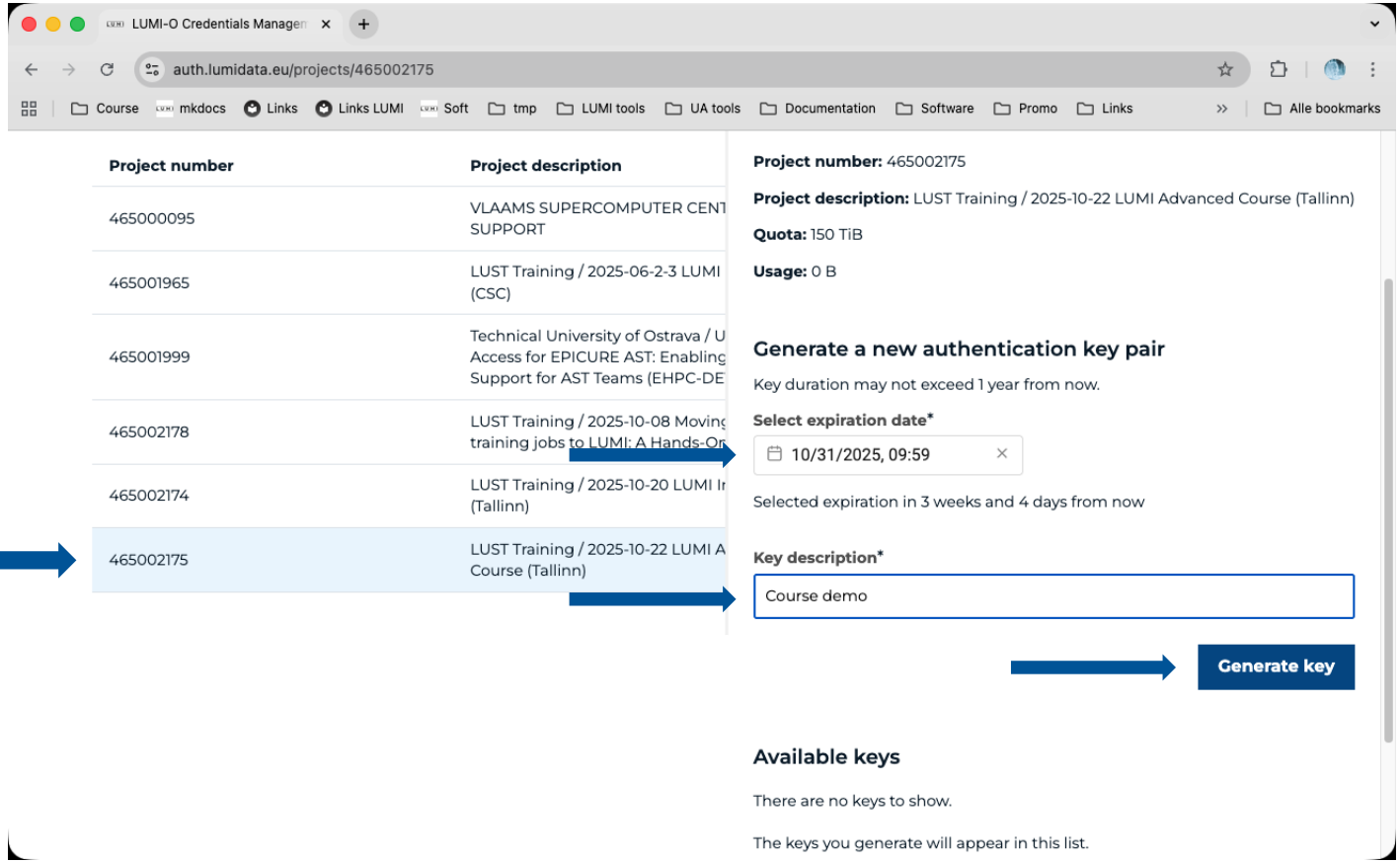


LUMI

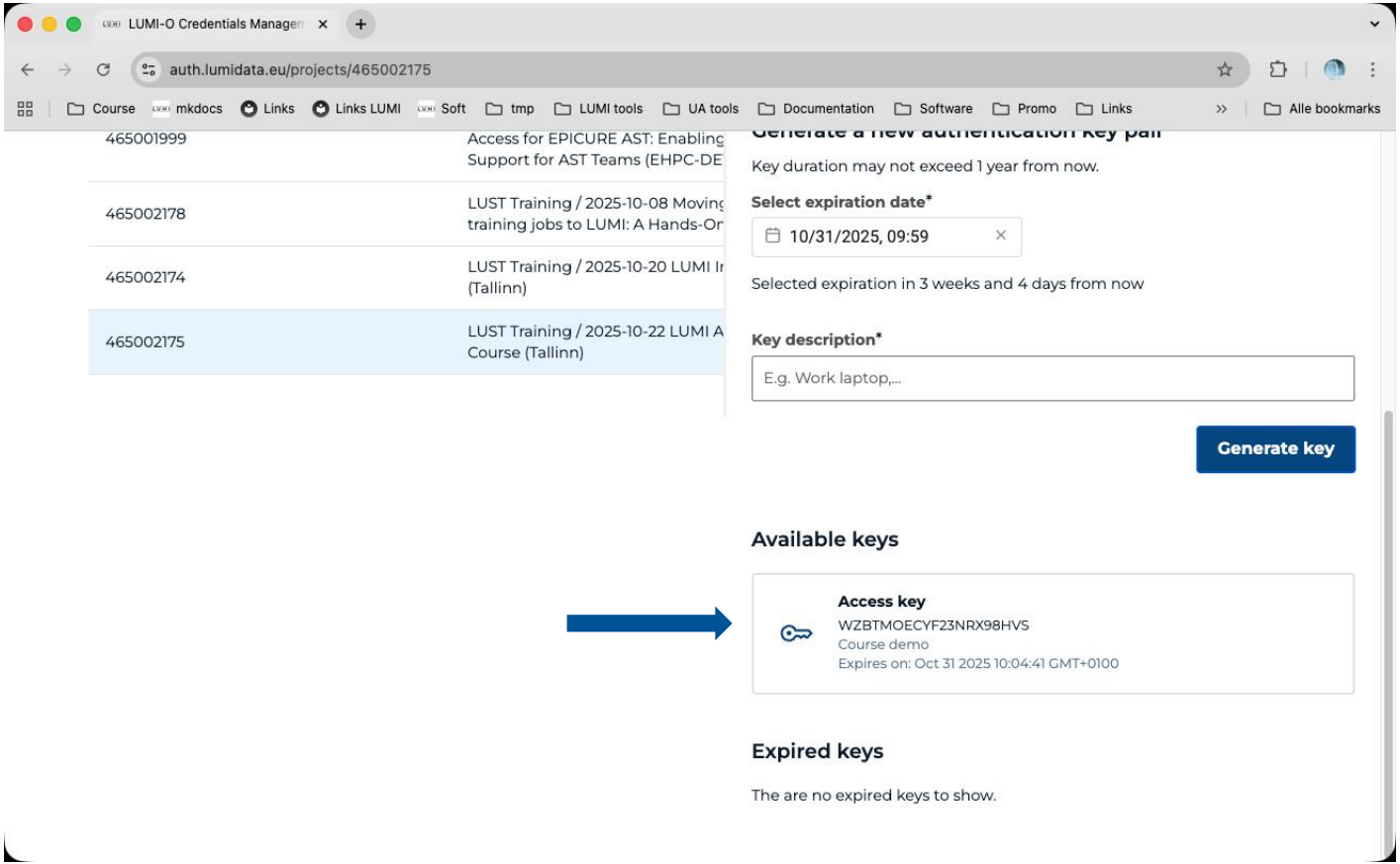
The credential management web interface

- Credential management web interface at auth.lumidata.eu
  - Create keys
  - Extend lifetime of a key (not needed that frequently anymore)
  - Create configure scripts for various tools
- You'll have to select your login method in the same way as for Open OnDemand
- After a while you should see a list of projects, select the one for which you want to generate a key
  - The right column will also show active keys for the project, and expired ones

Credential mangement web interface: Create credentials (3)



Credential mangement web interface: Create credentials (4)

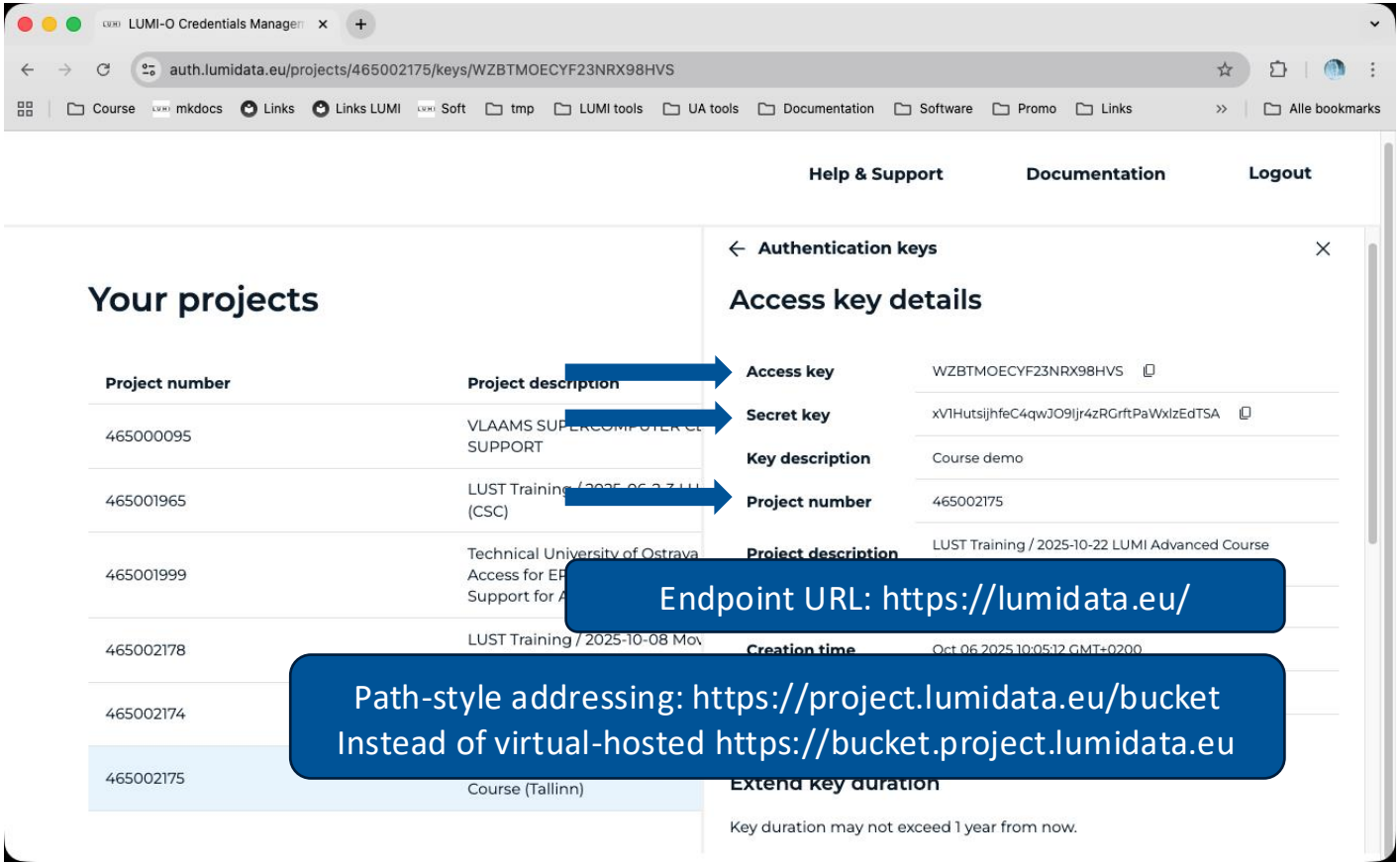


LUMI

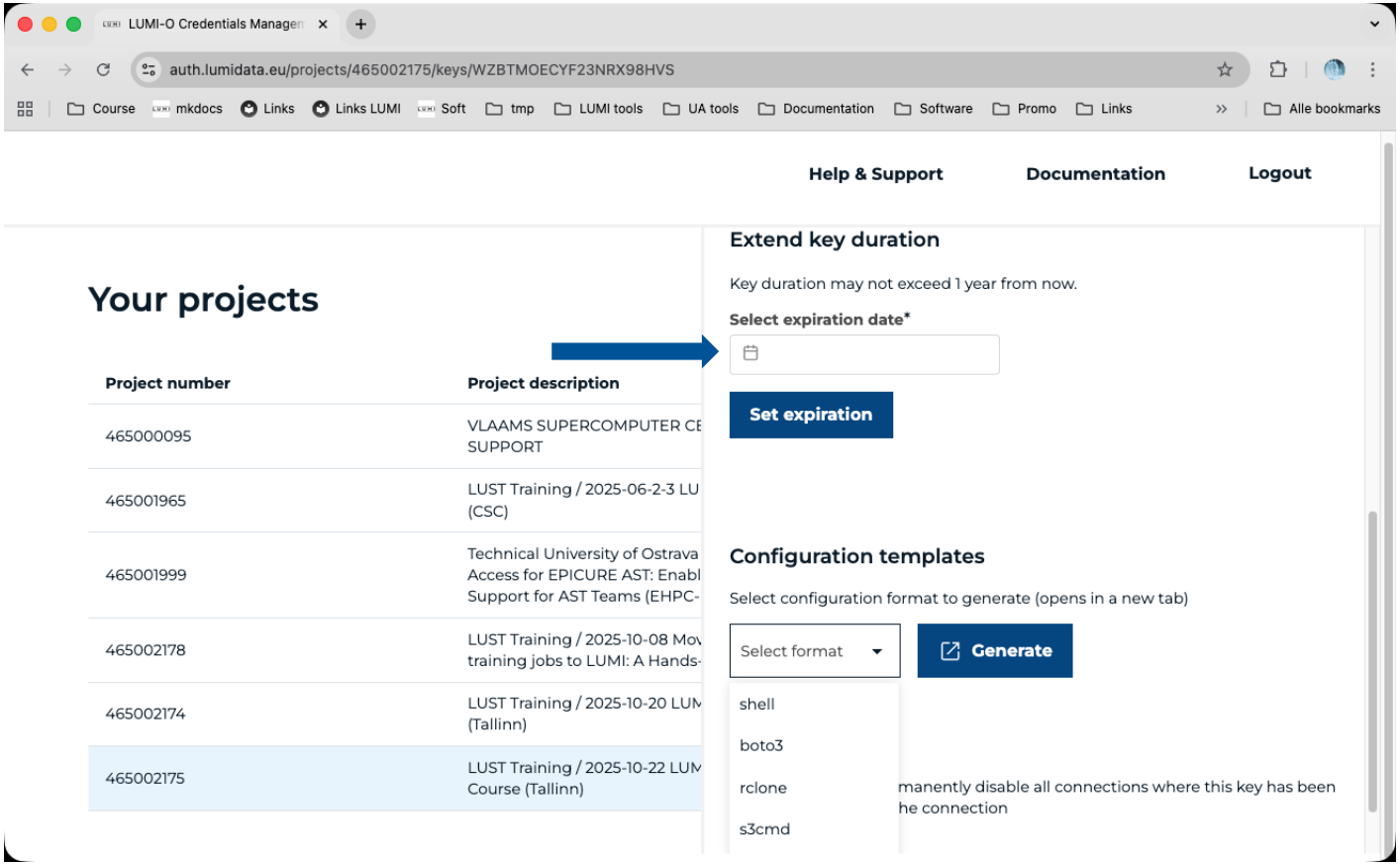
The credential management web interface

- Credential management web interface at auth.lumidata.eu
  - Create keys
  - Extend lifetime of a key (not needed that frequently anymore)
  - Create configure scripts for various tools
  - Often available during system maintenance also
- You'll have to select your login method in the same way as for Open OnDemand
- After a while you should see a list of projects, select the one for which you want to generate a key
  - The right column will also show active keys for the project, and expired ones
- Selecting an active access key changes the right column to one where you get information about the key, can extend the key duration and can generate templates to configure various tools

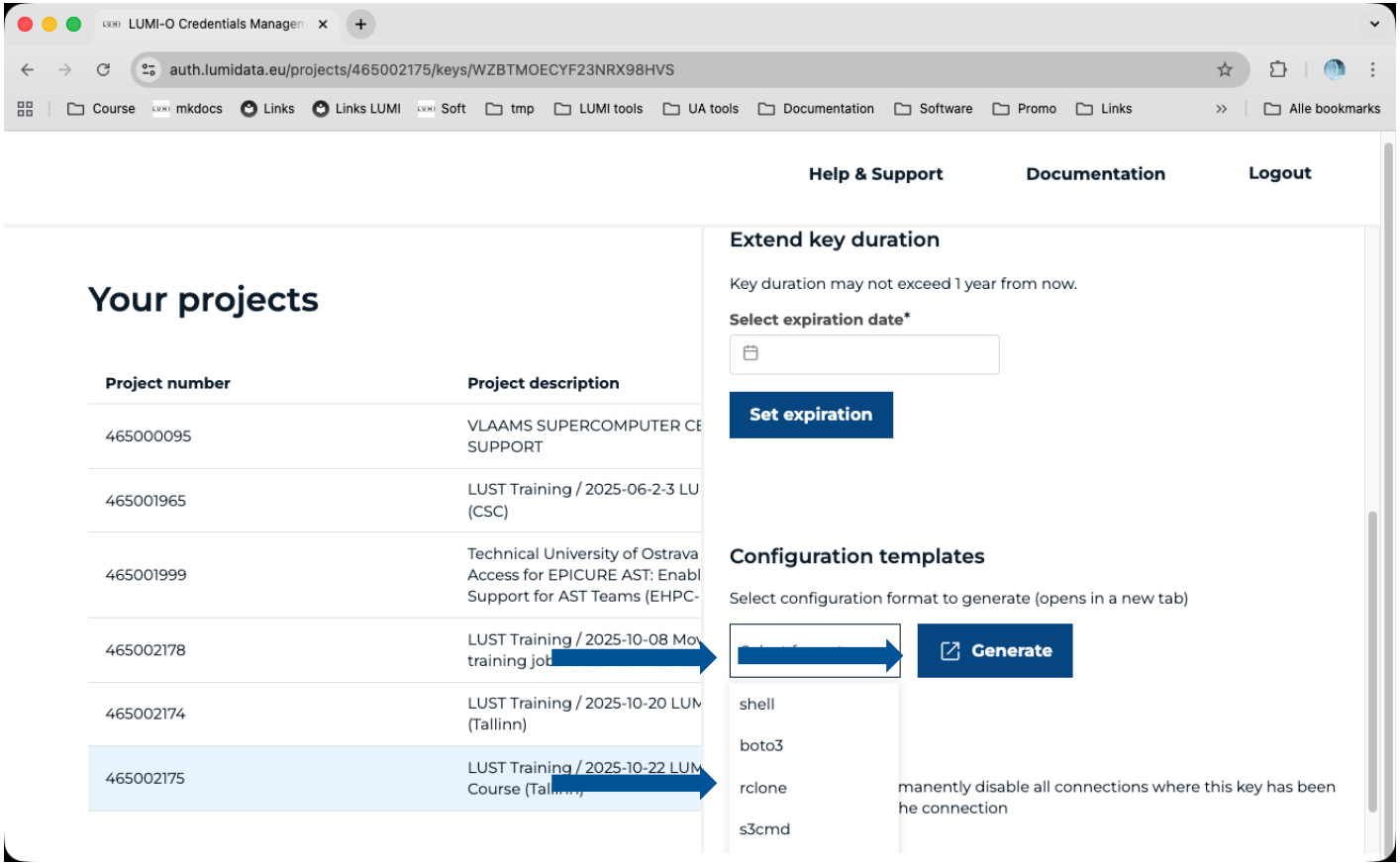
Credential mangement web interface: Check credentials



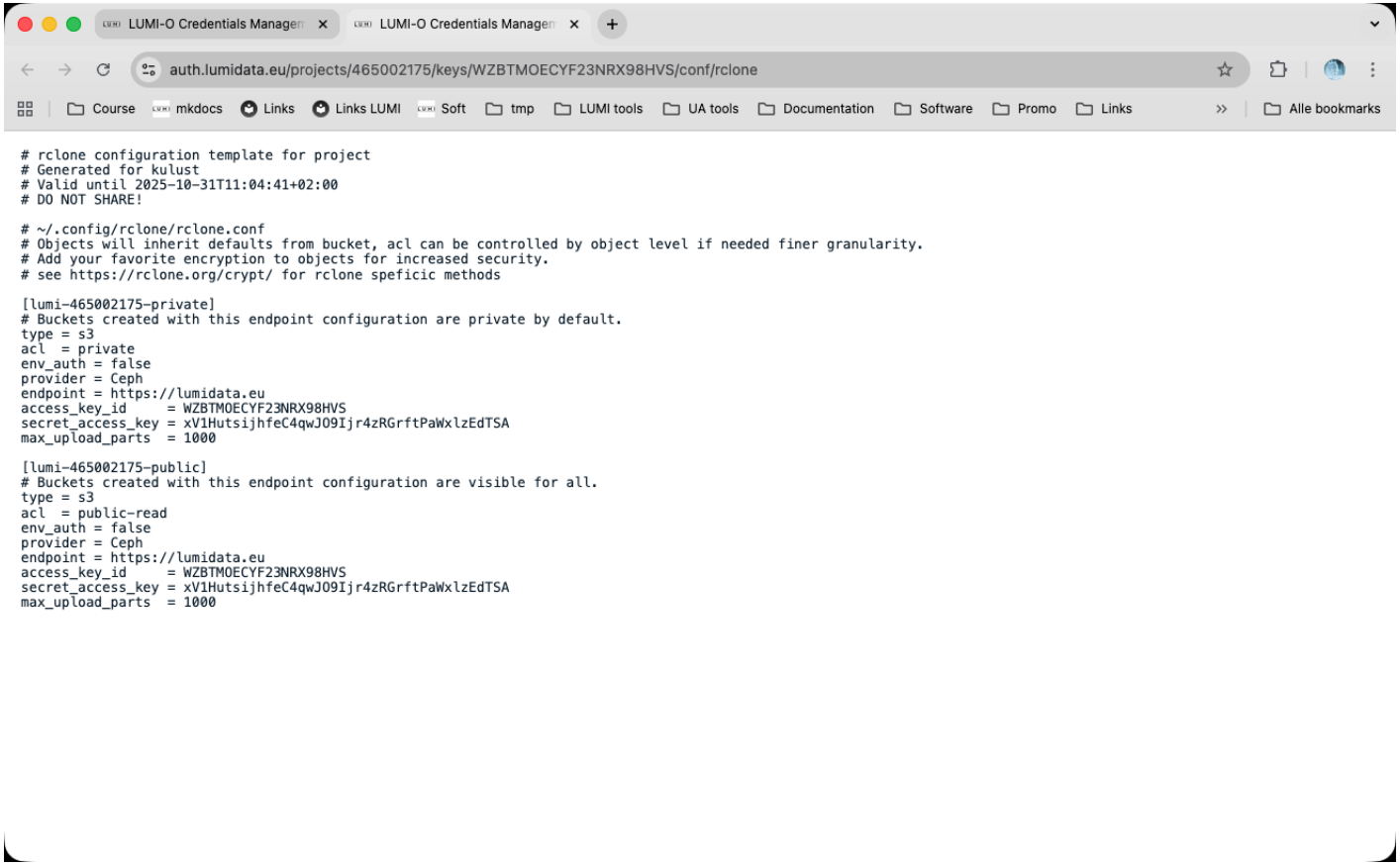
Credential mangement web interface: Extend credential lifetime



Credential mangement web interface: Tool configuration (1)



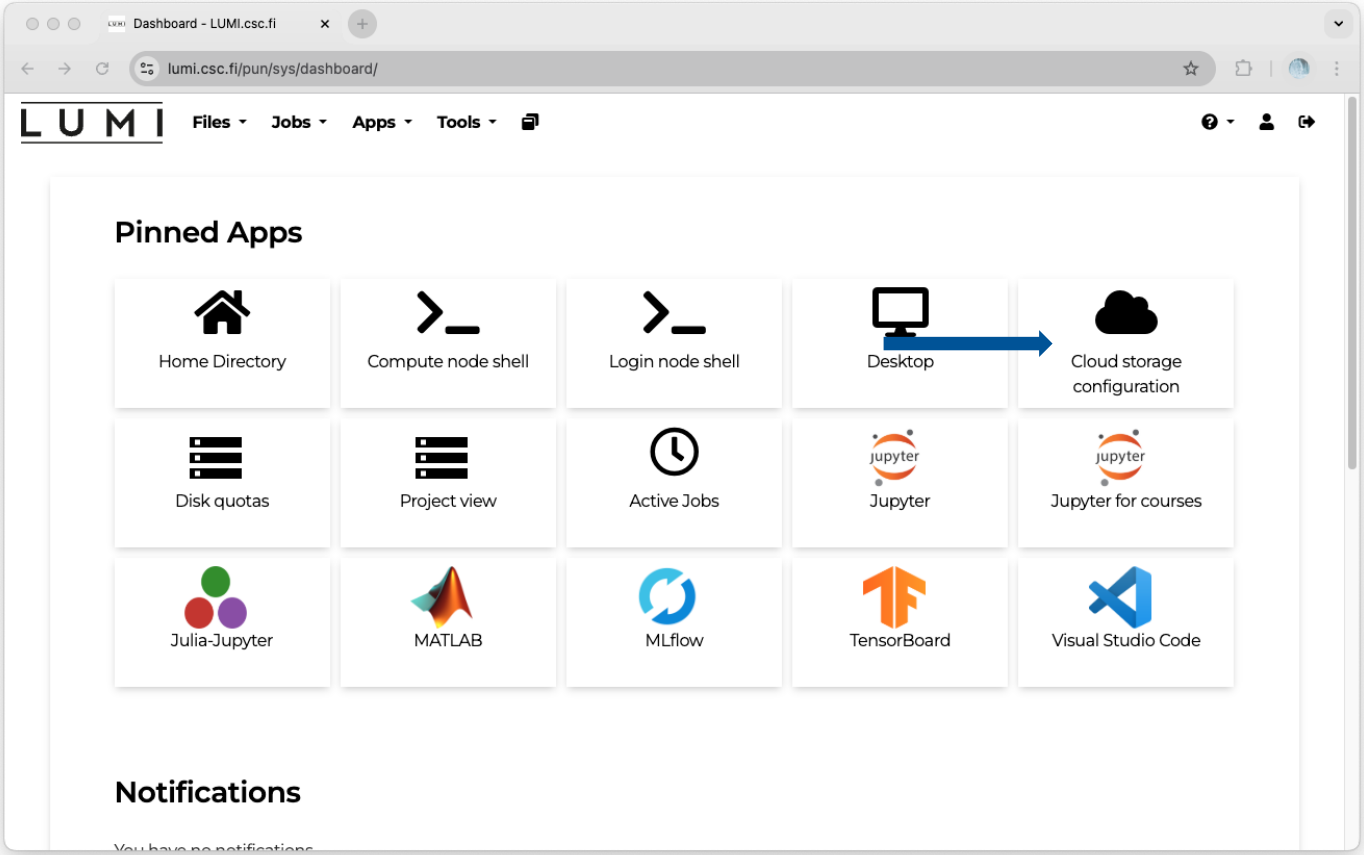
Credential mangement web interface: Tool configuration (2)



Access through Open OnDemand

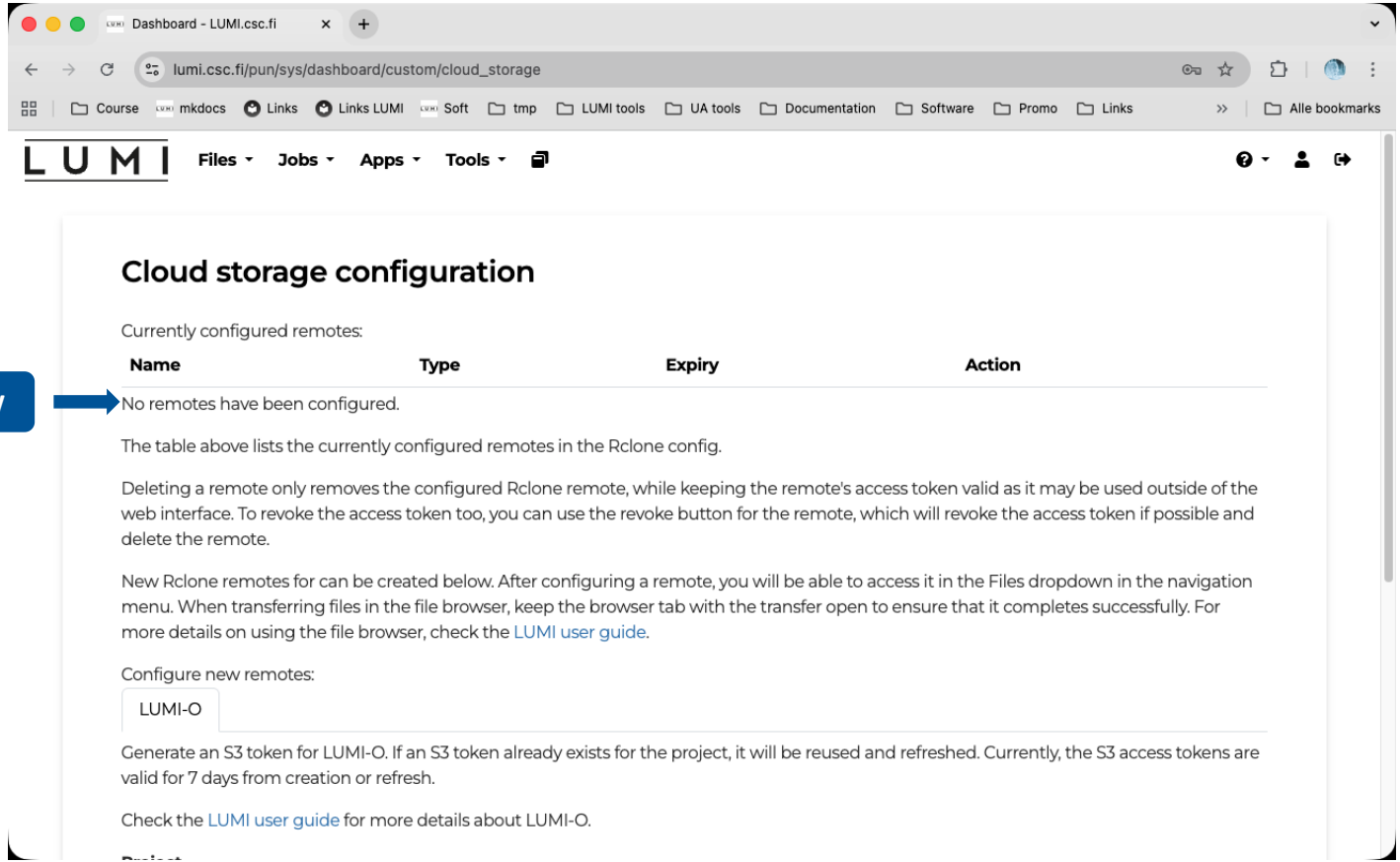
- Open OnDemand provides 2 apps:
  - An app for simple **credential management,** including generation of config files for `rclone` and `s3cmd` on LUMI.
  - The **"Home Directory" app** can be used to browse, download and upload objects and to create buckets.
    Structure resembles view of a typical rclone-based browser.
- Note that Open OnDemand is not a replacement for proper tools for access to LUMI-O!
  - Uploading and downloading is through web protocols that are not as performant as proper object storage tools.
- Open OnDemand is down during LUMI system maintenance!

Credential management via OOD (1)

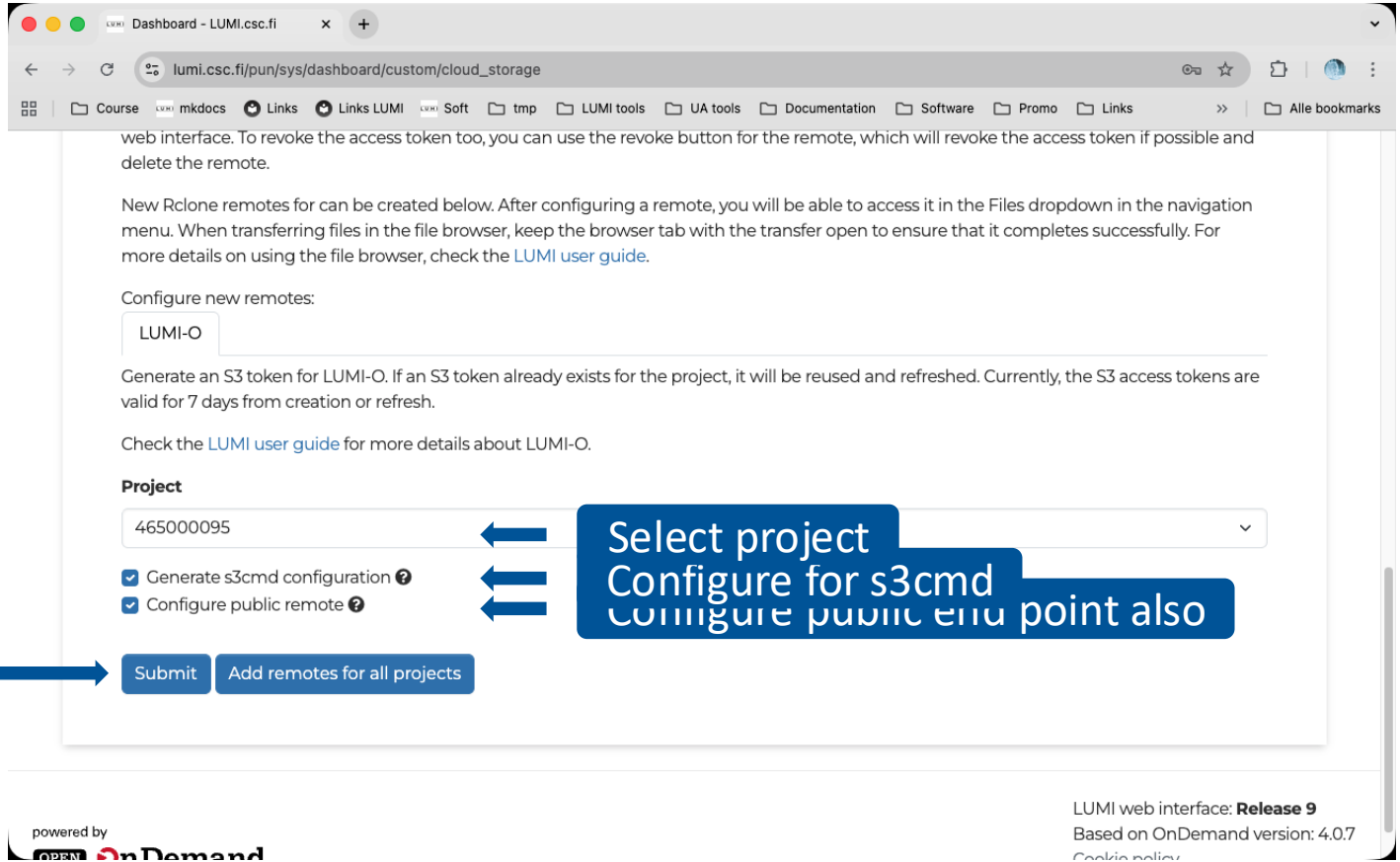


Credential management via OOD (2)

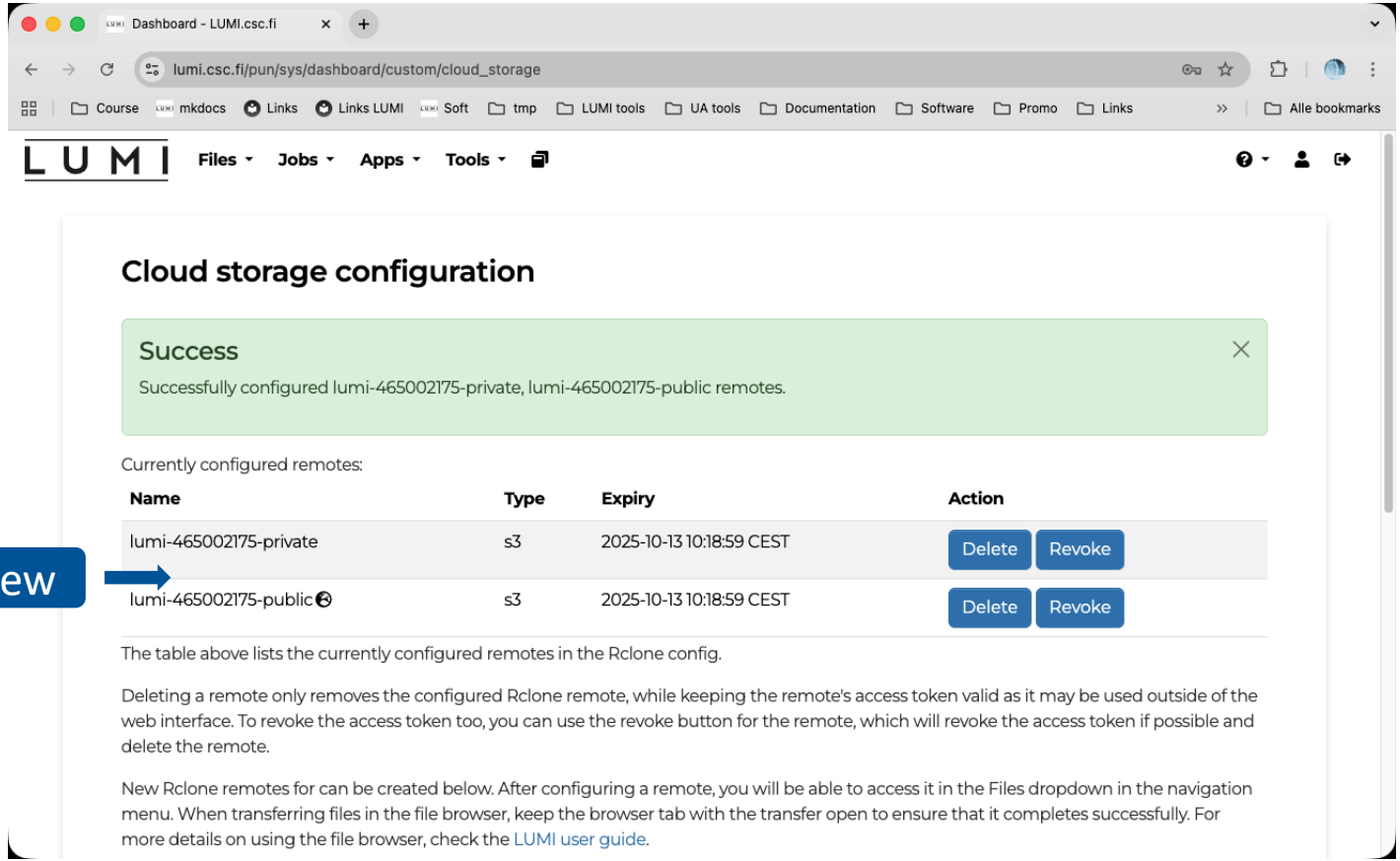
Overview



Credential management via OOD (3)

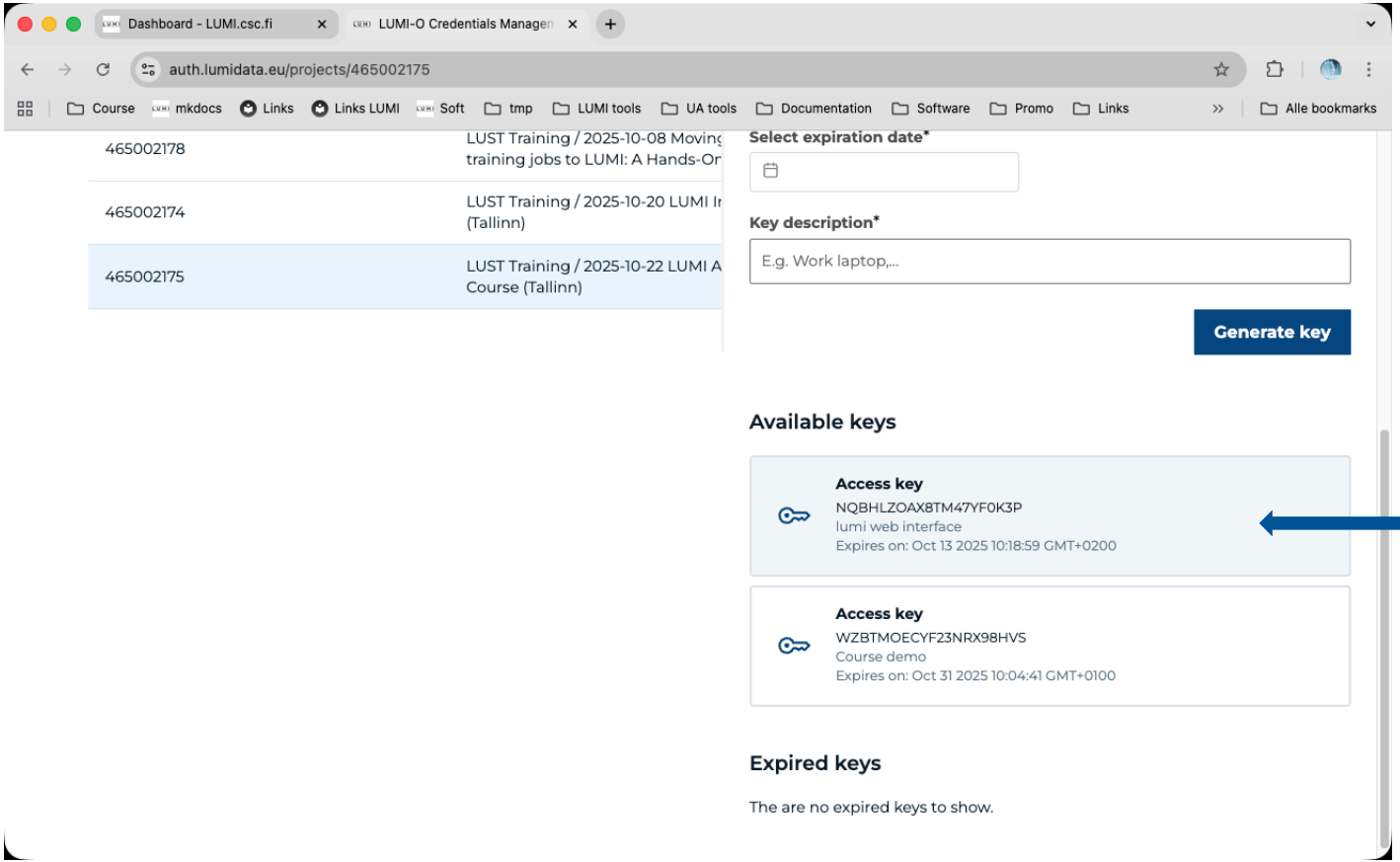


Credential management via OOD (4)

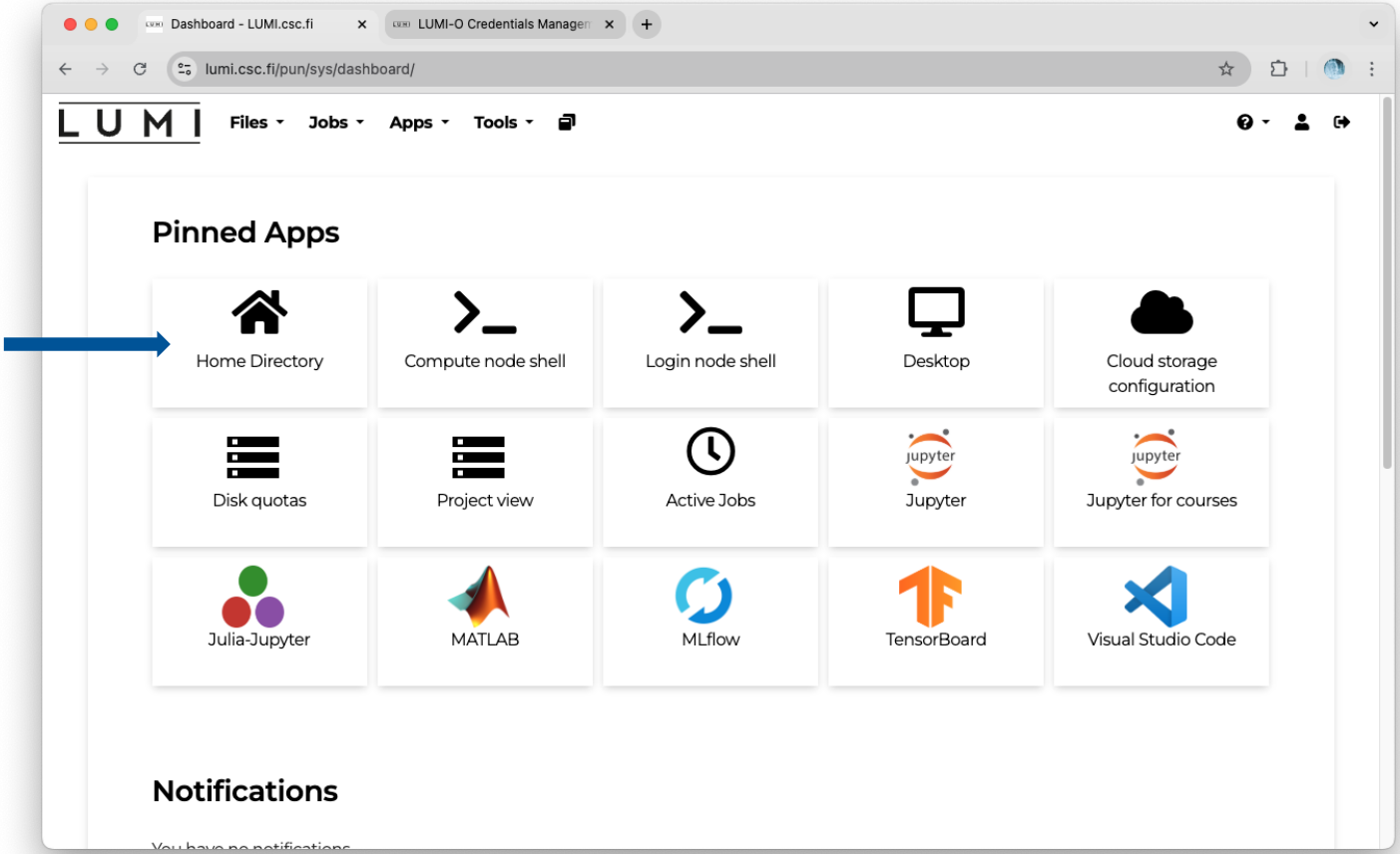


Overview

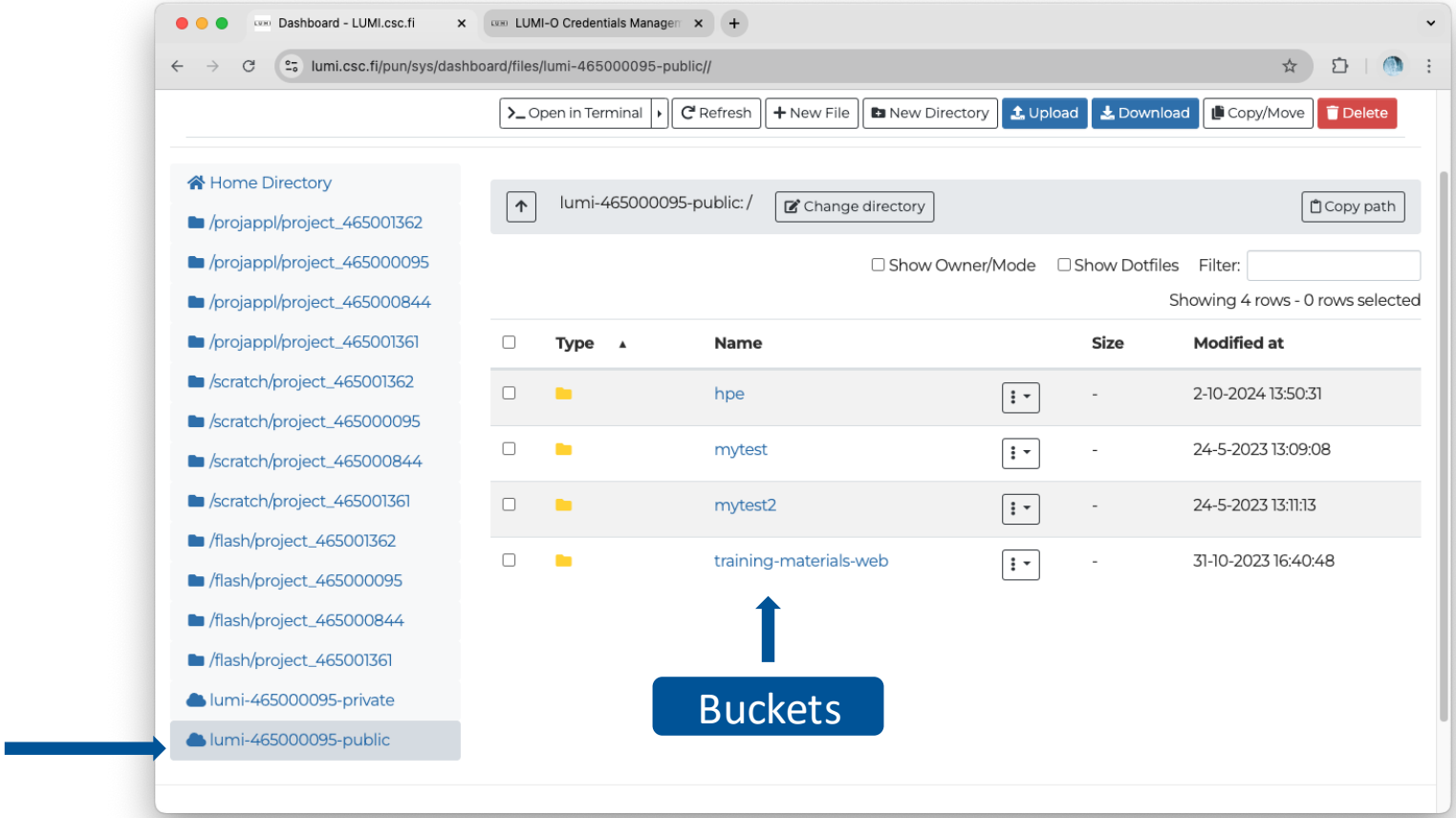
Credential management via OOD (5)



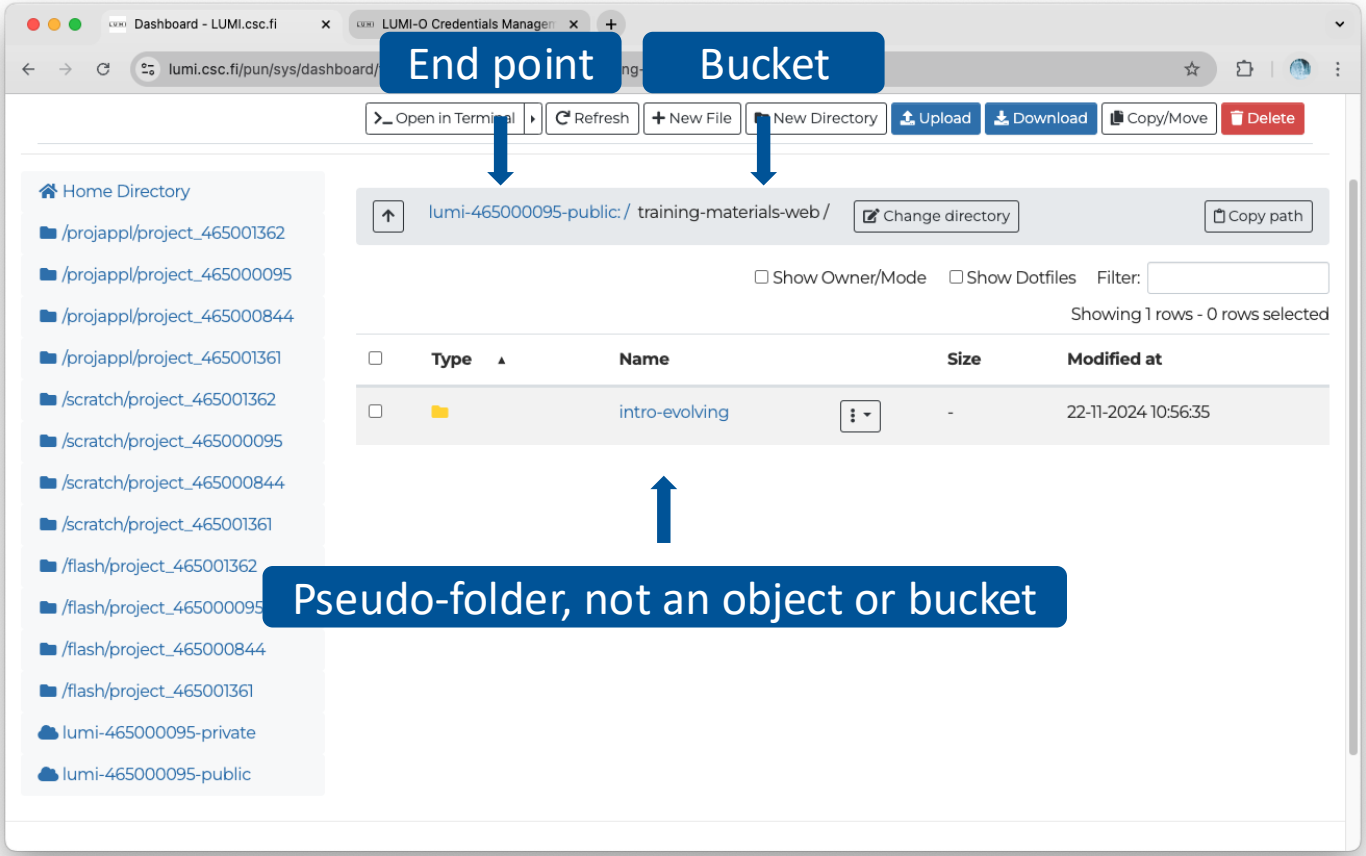
Browsing via OOD (1)



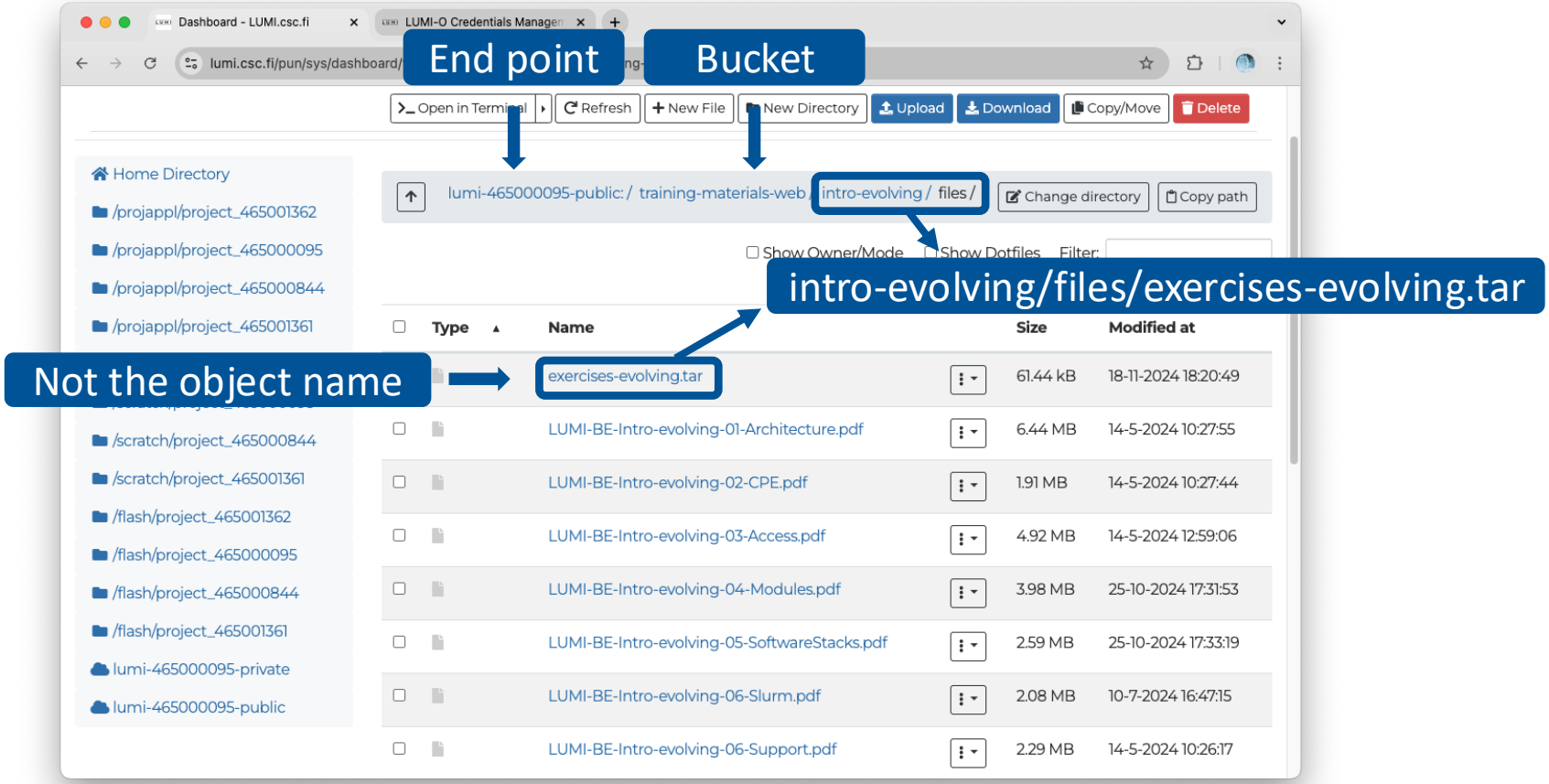
Browsing via OOD (2)



Browsing via OOD (3)



Browsing via OOD (4)



CLI tool configuration on LUMI: lumio-conf

LUMI

- On LUMI, you can use `lumio-conf` to configure `rclone`, `s3cmd`, `aws` and `boto3`
  - Need to load the `lumio` module which also provides `rclone`, `s3cmd` and `restic`
  - Will ask for data from the "Access key details" screen
  - Default is to create configurations for `rclone` and `s3cmd` .
    Try `lumio-conf -h` to see other options.
- Generate the configuration snippets via the web interface
  - E.g., for `rclone`: copy manually to `~/.config/rclone/rclone.conf`
  - Can be used to configure tools on your computer also.

CLI tool configuration on LUMI: lumio-conf, rclone and s3cmd

LUMI

- The `rclone` configuration contains two endpoints
  - `lumi-46YXXXXXX-private`: Private buckets and objects
  - `lumi-46YXXXXXX-public`: Public buckets and objects

  Note that this was different with an earlier version of the tool
- The `s3cmd` config file can only contain data for one project.
  Solution implemented by `lumio-conf`:
  - Store information for project 46YXXXXXX in `~/.s3cfg-lumi-46YXXXXXX`
  - Also overwrite `~/.s3cfg` with the same data
  - So s3cmd will by default use the project from `~/.s3cfg` but if a user has active configurations for multiple projects, the `-c` flag can be used to point `s3cmd` to the right configuration file.

LUMI

Bucket and object names

- Bucket name
  - Unique name within a project
  - Must be between 3 and 63 characters long, lowercase letters, numbers, hyphens and dots, must not contain uppercase characters or underscores
  - Must start with a lowercase letter or number
  - When dots are used, each part of the name that is separated by the dots, is also called a label (but there is a also a different concept of label in Ceph).
- Object name
  - Unique within a bucket
  - Technically, any UTF-8 string between 1 and 1024 bytes, but client software might add more restritions
  - Common practice is to implement a folder-like structure using slashes in the name.
  - But creating the folder view is expensive: List all objects in a bucket and sort

Policies and ACLs

- Access control is managed through bucket policies and bucket and object access control lists (ACLs)
- Policies is a very powerful but also hard to use mechanism
  - Some information in the "Advanced usage of LUMI-O" section of the docs
  - And there is also information in the Ceph manual
  - Can be managed through `s3cmd`
- ACLs apply to individual buckets and objects
  - Can only add rights
  - Useful to make a bucket or object public, or give access to another project, but this is done to individual objects (unless applied recursively)
  - This is what rclone uses when uploading to the public or private end points in the configurations created with the various tools for LUMI-O

Policies and ACLs Examples

LUMI

- Make a bucket and all objects in it public or private

```
s3cmd setacl --recursive --acl-public s3://bucket/
s3cmd setacl --recursive --acl-private s3://bucket/
```

- Grant or revoke read rights to a bucket, e.g., 46YXXXXXX giving rights to 46BAAAAAA:

```
s3cmd -c ~/.s3cfg-lumi-46YXXXXXX \
    setacl --acl-grant='read:46BAAAAAA$46BAAAAAA' s3://bucket
s3cmd -c ~/.s3cfg-lumi-46YXXXXXX \
    setacl --acl-revoke='read:46BAAAAAA$46BAAAAAA' s3://bucket
```

  - Note the use of single quotes to make sure that `$46BAAAAAA` is not interpreted as a variable name!
  - And similarly to objects

- Check the ACL and other information of a bucket or object

```
s3cmd info s3://2day-20261109
s3cmd info s3://2day-20261109/img/LUMI-2day-20261109-204-ObjectStorage/Title.png
```

Sharing data

- Public buckets and objects can be read by anyone in the world, even simply via a web browser.
- It is possible to grant specific projects access to buckets and objects.
  - See the previous slide for setting ACLs
  - Access: e.g., project "46BAAAAAA" can list a bucket from project "46YXXXXXX" (assuming the latter has given sufficient rights to the former) with:

```
s3cmd -c ~/.s3cfg-lumi-46BAAAAAA ls s3://46YXXXXXX:bucket/
s3cmd -c ~/.s3cfg-lumi-46BAAAAAA ls --recursive s3://46YXXXXXX:bucket/
rclone ls lumi-46BAAAAAA-private:"46YXXXXXX:bucket"
```

- Presigned URLs: Next slide
- Invite the other to the project
  - But it is not possible to add a user who came in via puhuri to a 462-project

Sharing data: Presigned URLs

- Can be generated using `rclone`
- Access to anyone who gets the link without further authentication, for the validity period of the link
- Presigned URLs depend on the authentication key that was used to create them.
  - Link expires when the key expires or is revoked, even if the validity period of the link has not expired.
  - Maximum lifespan of a link is limited to 7 days on LUMI-O.
- Create: with `rclone link`:

```
rclone link lumi-46YXXXXXX-private:bucket/object
rclone link --expire 2d lumi-46YXXXXXX-private:bucket/object
```

Some tips & tricks

- When using the `rclone` command line tool, it is possible to throttle the speed for many commands with the `--bwlimit` command line option
  - May be needed if you upload from home over a very bandwidth-limited connection
- Sharing data from project A with project B does not protect the data from being deleted when project A ends, even if project B is still valid.
  - End of project: 90-days grace period, ~~data read-only~~
  - End of grace period: Data queued for deletion


LUMI
Questions?